

## РОБОТЫ LEGO WEDO. ЗАНЯТИЕ 4. МАШИНКА С ДАТЧИКОМ РАССТОЯНИЯ

На этом занятии мы сосредоточимся на датчике расстояния, рассмотрим возможности программы Lego WeDo Education Software и программы Скретч по работе с датчиком расстояния, соберем машинку, которая будет оснащена одним, а потом двумя датчиками расстояния. Машинка будет ездить, обнаруживать препятствия и демонстрировать весьма сложное поведение.

### ДАТЧИК РАССТОЯНИЯ

Итак, объект нашего пристального внимания – датчик расстояния, изображенный на рис. 1.

Датчик оснащен двумя лампочками-глазками и проводом для подключения к USB ЛЕГО-коммутатору. Датчик обнаруживает предметы, находящиеся от него на расстоянии от 0 до 15 сантиметров. Подключенный датчик отображается в программе Lego WeDo Education Software на вкладке Связь, причем чем ближе к датчику находится препятствие, тем больше полосок занимает желтый индикатор (рис. 2).

Простейшее задание для датчика расстояния выглядит, например, как показано на рис. 3: при срабатывании датчика расстояния программа должна издать звук 1.

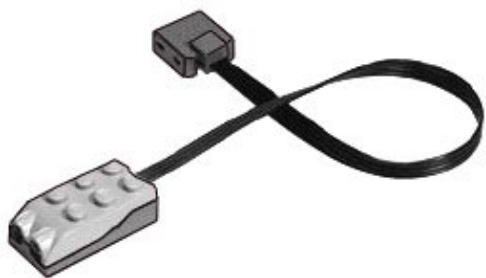


Рис. 1

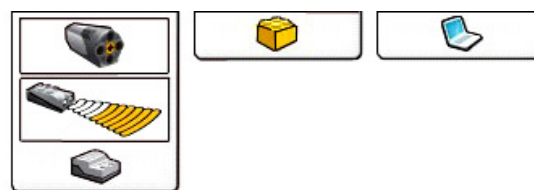


Рис. 2



Рис. 3.



Рис. 4

торое выдаст программа на рис. 4, а затем, не меняя положения датчика, положить черный лист. Значение изменится!

То же самое можно увидеть в программе Скретч. Подключенный датчик можно увидеть, если в группе Сенсоры выбрать сенсор «расстояние» и поставить флажок для отображения значения на экране (рис. 5).

Если двигать рукой перед датчиком, можно заметить, что значения меняются от 0 до 100, причем здесь наибольшее значение появляется на больших расстояниях, а ноль - при приближении к датчику. Заметно, что область видения датчика в Скретче больше, чем в Lego WeDo Education Software, однако значение от белого и черного листов также отличаются, то есть это датчик освещенности.

Программа, аналогичная изображенной на рис. 3, в среде Скретч будет выглядеть сложнее, так как команды «ждать датчика» здесь нет.

В Скретче, как показано на рис. 6, надо явно задавать то «расстояние», при котором срабатывает датчик. В данном случае указано «расстояние» 50. Положительным моментом можно отметить открывающуюся в

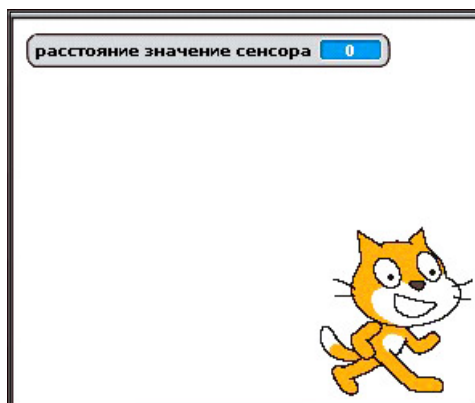


Рис. 5

Скретче возможность тонкого управления датчиком. Можно указать, чтобы датчик срабатывал только на малых расстояниях или срабатывал на расстояниях, соответствующих определенным дополнительным условиям.

### СБОРКА МАШИНКИ

Соберем робота-машинку и оснастим ее датчиком расстояния. Блочная инструкция по сборке машинки показана на рис. 7 а-и. В результате должна получиться машинка, показанная на рис. 7 л.

В машинке использованы детали только одного конструктора, поэтому коронная передача сделана из двух коронных колес, а большие шестеренки служат задними колесами. В отличие от модели из предыдущего номера, эта машинка не может поворачивать.

### ПРОГРАММИРУЕМ В СРЕДЕ LEGO WEDO EDUCATION SOFTWARE

После того как машинка собрана, надо заставить ее двигаться, обнаруживать препятствия и реагировать на них. Это задание может быть дано как творческое, каждый волон придумывать поведение для машинки в соответствии со своим замыслом и умениями программировать. Здесь мы предложим несколько решений задач, простых и не очень.

1. Пусть машинка едет вперед, пока не увидит препятствие. При срабатывании датчика расстояния программа должна воспроизвести звук 20 и отъехать назад, двигаясь 3 секунды.

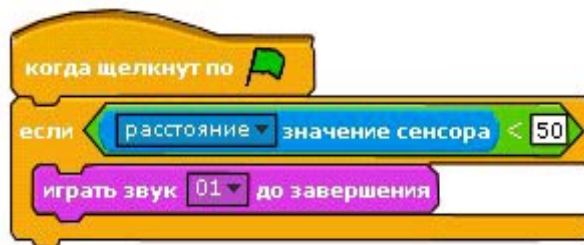


Рис. 6



Рис. 7



Рис. 8



Рис. 9



Рис. 10

Соответствующая программа показана на рис. 8.

Интересные возможности для управления дает использование пиктограммы Мощность мотора с входом Датчик расстояния. При запуске программы, показанной на рис. 9, на экране будет отображаться число, передаваемое датчиком расстояния при приближении или удалении от препятствия, при этом мотор будет крутиться также на разном уровне мощности.

При испытании программы хорошо заметно, что мотор крутится тем быстрее, чем ближе к «стенке» находится машинка. Хотелось бы, чтобы все было ровно наоборот: на большом расстоянии от стенки машинка должна ехать быстро (мотор должен рабо-

тать на мощности 10), а при приближении к стенке машинка должна останавливаться (мощность должна уменьшаться до нуля). Этого результата можно достичь вычитанием из 10 значения от датчика расстояния и передачей полученной разности на вход команды Мощность мотора (рис. 10).

Теперь машинка будет тормозить при приближении к стенке.

2. Теперь усложним программу, сделаем так, чтобы машинка периодически подъезжала, затормаживая, к стенке, а затем отъезжала от нее. Одно из возможных решений показано на рис. 11.

Программа стартует при нажатии кнопки В и сразу же отправляет письмо с текстом «пуск». При получении этого письма включается основная программа, определяющая движение машинки к стенке. Программа воспроизводит звук 1, включает мотор на мощности 10 и запускает цикл, в котором мощность мотора уменьшается при приближении к стенке. При падении мощности до нуля происходит выход из цикла, текст отправляемого письма принимает значение 0, то есть совпадает с текстом принимаемого письма в третьей части программы. Таким образом, после остановки машинки перед стенкой, запускается третья программа, в которой раздается звук 11, машинка на мощ-



Рис. 11

ности 10 начинает двигаться в противоположную сторону в течение двух секунд. После остановки снова отправляется письмо с текстом «пуск», стартует вторая часть, машинка снова движется к стенке.

Процесс повторяется, пока не будет нажата кнопка стоп или клавиша Esc на клавиатуре.

Программа Lego WeDo Education Software позволяет подключать несколько моторов и датчиков одновременно. Можно подключить второй датчик ко второму ЛЕГО-коммутатору и установить датчик на машинку так, чтобы он смотрел назад. Базовая программа по управлению такой машинкой показана на рис. 12.

Согласно этой программе машинка будет ехать вперед, пока не сработает датчик, смотрящий вперед, после этого она будет ехать назад, пока не сработает датчик, смотрящий назад. То есть можно, взмахивая руками перед датчиками, «гонять» машинку вперед-назад. Датчики различаются маркировкой, которую можно ставить и снимать, щелкая по пиктограмме при нажатой клавише Shift.

Программы, показанные на рис. 11 и 12, демонстрируют предельные возможности среды Lego WeDo Education Software. Дальнейшего усложнения поведения робота можно достигнуть, перейдя в среду программирования Скретч. В предыдущей статье мы познакомились со Скретчем на примере гото-



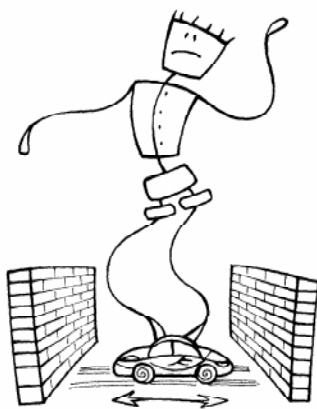
Рис. 12

вых проектов, а также программ для базовых моделей Lego WeDo. В этой статье будет приведено несколько примеров для машинки с датчиком расстояния, показывающих возможности работы и направления совершенствования.

### ПРОГРАММИРУЕМ В СРЕДЕ СКРЕТЧ

Как уже отмечалось, в среде Скретч необходимо явно указывать «расстояние», при котором происходит срабатывание датчика (см. рис. 6). Кроме условия срабатывания датчика, значение, получаемое от датчика расстояния можно использовать в целом множестве команд (плыть ... секунд, идти... шагов, повторять ... раз и т. д.), словом, везде, где нужно указывать число.

К примеру, программа, показанная на рис. 13, заставляет машинку с одним датчиком расстояния двигаться вперед, издавая короткие гудки, обнаруживать «стенку», отъезжать назад в течение трех секунд и снова двигаться к стенке.



*...машинка будет ехать вперед, пока не сработает датчик, смотрящий вперед, после этого она будет ехать назад, пока не сработает датчик, смотрящий назад.*



Рис. 13

Здесь, в отличие от программы, показанной на рис. 11, не требуется отсылать и получать сообщения, чтобы заставить машинку двигаться в бесконечном цикле; в Скретче управляющая конструкция «если – или» позволяет сделать это проще. «Но ведь машинка не тормозит!» – скажете вы и будете совершенно правы. В Скретче можно легко изменять мощность мотора, сделать ее зависимой от значения датчика расстояния.

Рассмотрим задачу: используя мотор и датчик расстояния, написать в среде Скретч программу, согласно которой мощность мотора зависит от расстояния до препятствия. В программе должен меняться звук при приближении и удалении от препятствия. Одно из возможных решений приведено на рис. 14.

Здесь созданы две переменные  $a$  и  $b$ , каждой из которых с задержкой в 0,1 секунды присвоено значение, получаемое от датчика расстояния. Это сделано для того, чтобы программа сама определила, удаляется машинка от препятствия или приближается к нему. Звук будет разным, в зависимости от того, увеличилось или уменьшилось расстояние за десятую долю секунды. Мощности мотора присваивается значение  $a + 20$ , то есть мощность меняется пропорционально числу, выданному датчиком расстояния. Это называется пропорциональным регулятором.

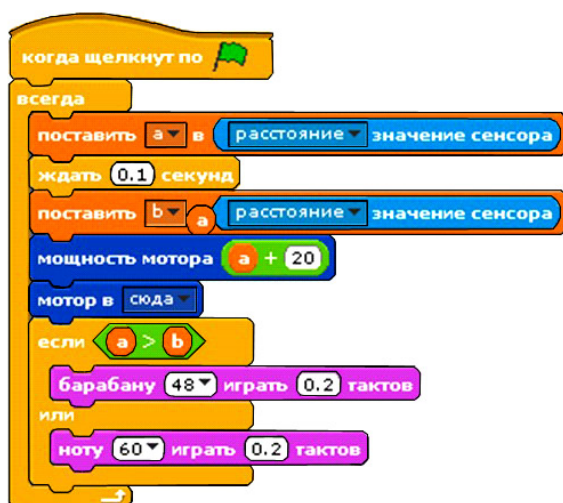
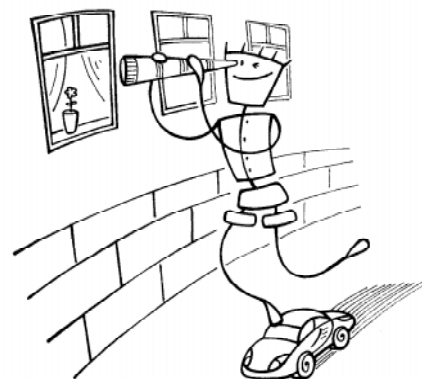


Рис. 14

Среда Скретч является более мощной, а следовательно, более гибкой, позволяет ставить и решать самые разнообразные задачи по управлению. Ограничения этой среды связаны с невозможностью (по крайней мере, в версии 1.4) использовать несколько моторов и датчиков. Если бы Скретч поддерживал управление хотя бы двумя моторами, можно было бы собрать машинку с двумя моторами (см. предыдущую статью), оснастить ее датчиком расстояния и запрограммировать машинку на прохождение лабиринта. Машинку с одним мотором можно запрограммировать на псевдопрохождение лабиринта: можно сделать, чтобы машинка ехала вперед, а при обнаружении препятствия на экране появлялась бы надпись с просьбой повернуть машинку, например на 90 градусов вправо. Далее программа проверяла бы наличие препятствия перед датчиком и либо отправляла машинку вперед, если препятствия нет, либо снова просила выполнить поворот.

В заключение этого занятия мы разберем еще одну робототехническую задачу: организовать движение машинки вдоль стенки с окошками и подсчет окошек. Вариацией на тему может стать задача подсчета черных полос на белом фоне. Поскольку наша машинка не умеет поворачивать, будем считать, что стенка прямая, развернем датчик расстояния на 90 градусов, так, чтобы он смотрел на «стенку», а стенку можно изготовить из нескольких коробок или книжек, установленных вдоль прямой линии на некотором



...организовать движение машинки вдоль стенки с окошками и подсчет окошек.

расстоянии друг от друга. Одно из бесчисленного множества возможных решений представлено на рис. 15.

Здесь созданы три переменные. Первая «*a*» исключительно для удобства принимает значения, передаваемые датчиком расстояния. Вторая «окно» введена для подсчета количества обнаруженных окон. А третья переменная «начало» (начало окна) принимает два значения 0 и 1, она служит флагом для разделения движения машинки вдоль стенки (0) и вдоль окна (1). Оказываясь перед окном, датчик расстояния показывает большие значения, присваиваемые переменной *a*. Если эти значения больше, например, 80 и флаг «начало» имеет значение 0 (то есть до того датчик видел стенку), программа говорит, что обнаружено новое окошко, увеличивает счетчик окон на единицу и ставит флаг в значение 1 (то есть датчик видит окно). Следующее условие проверяет, не сменилось ли окно стенкой. Если расстояние уменьшилось ( $a < 80$ ) и флаг «начало» = 1 (то есть до того датчик видел окно), ставим флаг в значение 0, символизирующее конец окна.

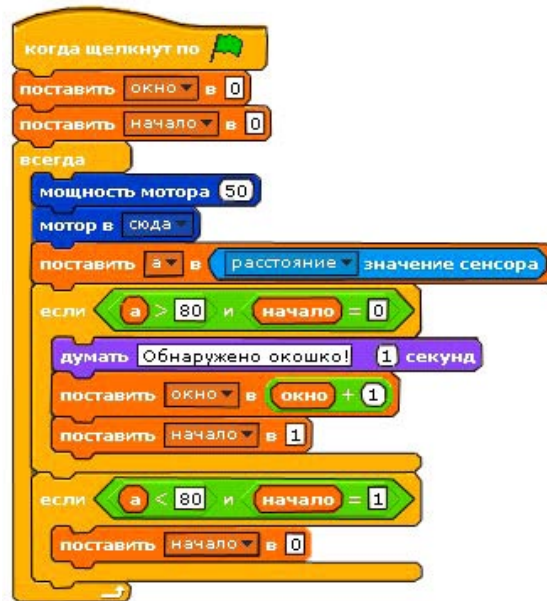


Рис. 15

Таким образом, мы в достаточно полной мере изучили использование датчика расстояния в задачах по обнаружению препятствий, причем сделали это в среде Lego WeDo Education Software и в среде Скретч.

*Порохова Ирина Алексеевна,  
кандидат физико-математических наук,  
заведующая Образовательного центра «ИНТОКС».*



Наши авторы, 2012.  
Our authors, 2012.